

VIII.3.3-RES-SNGL-PO2 SINGLE RESERVOIR REGULATION OPERATION  
RESERVOIR COMMAND LANGUAGE (RCL) STORAGE  
FORMAT

The Reservoir Command Language (RCL) for Operation RES-SNGL is stored in the PO array in an encoded numerical form.

The pointer to the starting in the PO array of the RCL information is held in word 12 of the PO array.

The 6 major RCL code groups and their numerical ranges are:

<u>Group</u>	<u>Numerical Range</u>
1. RCL keywords	1 - 50
2. Relational operators	101 - 200 (see Section VIII-3.3-RES-SNGL-PO3)
3. System variables	501 - 550 (see Section VIII-3.3-RES-SNGL-PO3)
4. Logical variables	551 - 600 (see Section VIII-3.3-RES-SNGL-PO3)
5. System functions	601 - 700 (see Section VIII-3.3-RES-SNGL-PO4)
6. Schemes and utilities	1001 - 2000

More details for groups 2, 3, 4 and 5 are provided in Sections VIII.3.3-RES-SNGL-PO3 and VIII.3.3-RES-SNGL-PO4.

If IF statements are used in the RCL, the codes for the IF expressions are stored in separate sections of the PO array and only references to their locations are made from this section of the PO array.

The keywords allowed within the RCL and their individual codes are:

<u>RCL Keyword</u>	<u>Numerical Code</u>
1. SET	1.01
2. DO	5.01
3. IF	10.01
4. THEN	11.01
5. ELSE	12.01
6. ELSEIF	13.01
7. ENDIF	19.01
8. ENDRCL	9.01

The codes (both here and elsewhere) are appended with a 0.01 to ensure that conversion from a real variable (as held in the PO array) to an integer does not result in truncation to a value one less than expected.

Codes have been assigned to both the SET and THEN keywords, but these values do not appear within the PO array. Their use and location are implied by the surrounding RCL keywords and structure.

The format of the PO array following any of the above keywords depends on which keyword is found. If the keyword is a DO, the code

for the keyword is always followed by the code for the scheme or utility that is to be executed. Not all utilities can be used with a DO and any attempt to use one that cannot will be screened and marked as an error when the operation is defined. The following table lists all the scheme and utility identifiers as well as their code numbers and whether they are allowed with the DO keyword. (The Scheme/Utility code numbers are the base numbers. The actual number stored for any Scheme/Utility is its base number plus its level of definition, since any scheme or DO-activated utility can be multiply defined).

<u>Scheme/Utility Identifier</u>	<u>Numerical Code</u>	<u>IBASE</u>	<u>DO Activated</u>
PASSFLOW	101X.01 <u>1</u> /	1	yes
SETQ	102X.01	2	yes
SETH	103X.01	3	yes
RULECURVE	104X.01	4	yes
FILLSPILL	105X.01	5	yes
SPILLWAY	106X.01	6	yes
POOLQ	107X.01	7	yes
STPOOLQ	108X.01	8	yes
MINQ	109X.01	9	yes
MINH	110X.01	10	yes
INDSRCHGE	111X.01	11	yes
FLASHBDS	112X.01	12	yes
POWERGEN	113X.01	13	yes
SETDQ	114X.01	14	yes
SETDH	115X.01	15	yes
RULEADJ	151X.01	51	no
SUMINF	152X.01	52	no
RAINEVAP	153X.01	53	no
ADJUST	154X.01	54	no
BACKFLOW	155X.01	55	no
MAXQ	157X.01	57	no <u>2</u> /
ENTERISC	158X.01	58	yes
SETMIN	159X.01	58	yes
SETMAX	160X.01	60	yes
GOFLASH	161X.01	61	yes

Notes:

1/ X is the level of definition of the Scheme/Utility.  $X \geq 1$ , default to 1.

2/ This utility is used within an IF expression and its use is discussed in Section VIII.3.3-RES-SNGL-PO4.

Following the Scheme/Utility code is either the next instruction or the location in the PO array to which to proceed. A positive value indicates an instruction whereas a negative value indicates a 'jump'. Consider the following RCL statements:

```
DO SETQ
ENDRCL
```

These statements are encoded in the PO array as:

<u>5.01</u>	<u>1021.01</u>	<u>9.01</u>
DO	SETQ	ENDRCL

In this example, the DO is not part of an IF statement group, hence the code following the SETQ code is positive. That code is then executed. Being part of an IF group is the situation that can result in the code following the Scheme/Utility code to be negative. Examples will be given in the discussion of the IF keyword to illustrate that situation.

Consider successive DO statements, as in:

```
DO STPOOLQ
DO INSDRCHGE
DO SETMIN
ENDRCL
```

Encoded, these statements appear as:

<u>5.01</u>	<u>1081.01</u>	<u>5.01</u>	<u>1111.01</u>	<u>5.01</u>	<u>1591.01</u>	<u>9.01</u>
DO	STPOOLQ	DO	INSDRCHGE	DO	SETMIN	ENDRCL

Once again, after each Scheme/Utility code, the code number is positive, no jump is necessary and the code is executed until the ENDRCL keyword code is reached. If the keyword is an IF, the code for the IF is followed by the location of the IF expression in the PO array, the location that RCL decoding is to proceed if the IF expression tests 'false' and then the action to take (a DO keyword) if the IF expression tests 'true'.

The IF expression location is denoted by a negative number. The absolute value of this number indicates how many words from the start of the IF expression section of the PO array this particular IF expression is located. The 'false' action location is also negative and the absolute value of this number indicates to where logical control passes. This indicator is also a relative pointer; it provides the offset from the start of the RCL information in the PO array. Following the 'false' transfer location is an encoded DO keyword followed by the code for the Scheme/Utility to be executed.

The following is a simple IF statement group:

```
IF (expression) THEN DO SETQ
ENDIF
ENDRCL
```

In the encoded form, this statement would appear as:

<u>10.01</u>	<u>-1.01</u>	<u>-6.01</u>	<u>5.01</u>	<u>1021.01</u>	<u>19.01</u>	<u>9.01</u>
IF	<u>1/</u>	<u>2/</u>	DO	SETQ	ENDIF	ENDRCL

Notes:

The IF keyword code starts in position 1 of the RCL section of the PO array.

- 1/ Information about (expression) begins at position 1 in IF EXPRESSION section of the PO array (see Section VIII.3.3-RES-SNGL-PO3).
- 2/ Go to position 6 of RCL section of PO array if (expression) tests false. In this case, control passes to the ENDIF keyword located in position 6. Control passes to the next word in the PO array if the (expression) tests true; here to word 5, the DO statement.

The structure of the PO array following an ELSEIF keyword is identical to that following an IF keyword except for the keyword code.

The IF group structure allows for either an alternative choice or an alternative action through the use of the ELSEIF and ELSE keywords, respectively.

The following examples will illustrate the use and storage layout of the ELSEIF keyword.

```
IF (expression 1) THEN DO SETQ
ELSEIF (expression 2) THEN DO SETH
ENDIF
ENDRCL
```

Encoded, these RCL statements look like this in the RCL section of PO array:

<u>10.01</u> IF	<u>-1.01</u> <u>1/</u>	<u>-7.01</u> <u>2/</u>	<u>5.01</u> DO	<u>1021.01</u> SETQ	<u>-12.01</u> <u>3/</u>		<u>13.01</u> ELSEIF	<u>-5.01</u> <u>4/</u>	<u>-12.01</u> <u>5/</u>	<u>5.01</u> DO	<u>1031.01</u> SETH	<u>19.01</u> ENDIF	<u>9.01</u> ENDRCL
--------------------	---------------------------	---------------------------	-------------------	------------------------	----------------------------	--	------------------------	---------------------------	----------------------------	-------------------	------------------------	-----------------------	-----------------------

Notes:

The IF keyword code starts in position 1 of the RCL section of the PO array.

- 1/ Information about (expression 1) begins at position 1 in IF EXPRESSION section of PO array.
- 2/ Go to position 7 of RCL section of PO array if (expression 1) tests 'false'. In this case, the ELSEIF test is to be made. The ELSEIF keyword code is in position 7.
- 3/ Go to position 12 of RCL section of PO array if the (expression 1) tests 'true'. In this case no other tests or actions are to be taken upon receipt of the 'true' test. In fact, after the specified actions are executed after a 'true' IF or ELSEIF expression test, control will always pass to the end of the IF statement group, the ENDIF keyword code. In this example, the ENDIF is located in position 12.
- 4/ Information about (expression 2) begins at position 5 in IF EXPRESSION section of the PO array (see Section VIII.3.3-RES-SNGL-

PO3).

- 5/ Go to position 12 of RCL section of PO array if IF (expression 2) tests 'false'. In this instance, control passes to the ENDIF keyword, located in position 12.

An ELSE keyword is always followed by a DO keyword. An example of the layout of the ELSE keyword is:

```
IF (expression 1) THEN DO SETQ
ELSE DO SETH
ENDIF
ENDRCL
```

These statements are encoded in the RCL section of PO array into:

```
10.01   -1.01   -7.01   5.01   1021.01   -10.01
IF       1/      2/      DO      SETQ      3/

12.01   5.01   1031.01  19.01   9.01
ELSE    DO      SETH      ENDIF    ENDRCL
```

Notes:

The IF keyword code starts in position 1 of the RCL section of the PO array.

- 1/ Information about 'expression 1' begins at position 1 in IF EXPRESSION section of the PO array (see Section VIII.3.3-RES-SNGL-PO3).
- 2/ Go to position 7 of RCL if (expression 1) tests 'false'. In this case, control passes to the ELSE keyword in position 7.
- 3/ Position 10 is the location of the ENDIF keyword. Go to position 10 of RCL section of PO array if (expression 1) tests 'true'.

In the explanation of the format for the DO keyword, it was mentioned that a negative value could follow the Scheme/Utility code indicating a processing 'jump'. The previous examples illustrating the layout for the ELSEIF and ELSE keywords have shown this situation.

In all the examples, simple situations have been presented to demonstrate the PO array structure for the various keywords. All these situations can be combined to form a legitimate arrangement of RCL statements. For example:

```
DO ENTERISC
IF (expression 1) THEN DO INDSRCHGE
ELSEIF (expression 2) THEN DO RULECURVE
    IF (expression 3) THEN DO SETQ
        IF (expression 4) THEN DO SETH
    ENDIF
ENDIF
ELSE DO PASSFLOW
```

```

DO STPOOLQ
DO SETMIN
ENDIF
ENDRCL

```

The PO array encoded form of these statements in the RCL section of PO array is:

```

5.01   1581.01   10.01   -1.01   -9.01   5.01   1111.01   -34.01
DO      ENTERISC   IF      1/      2/      DO      INDSRCHGE   3/

13.01   -5.01   -27.01   5.01   1041.01   10.01   -9.01   =
25.01
ELSEIF   4/      5/      DO      RULECURVE   IF      6/      7/

5.01   1021.01   10.01   -13.01   -24.01   5.01   1031.01   19.01
DO      SETQ      IF      8/      9/      DO      SETH      ENDIF

19.01   -34.01   12.01   5.01   1011.01   5.01   1081.01   5.01
ENDIF   10/    ELSE     DO      PASSFLOW   DO      STPOOLQ   DO

1591.01 19.01   9.01
SETMIN  ENDIF  ENDRCL

```

Explanation Notes:

The first DO keyword code starts in position 1 of the RCL section of the PO array.

- 1/ 4/ 6/ 8/ Beginning positions of (expressions 1, 2, 3 and 4) begin at 1, 5, 9 and 13 respectively, in the IF EXPRESSION section of the PO array (see Section VIII.3.3-RES-SNGL-PO3).
- 2/ Go to position 9 of the RCL section of the PO array if (expression 1) tests 'false'. Here control is to pass to the ELSEIF keyword in position 9.
- 3/ Go to position 34 of the RCL section of the PO array if 'true' test of (expression 1). Position 34 is the ENDIF keyword paired with the first IF keyword.
- 5/ Go to position 27 of the RCL section of the PO array if (expression 2) tests 'false'. Control passes to position 27 in this case, the ELSE keyword that is associated with the first IF keyword.
- 7/ Go to position 25 of the RCL section of PO array if (expression 3) tests 'false'. Position 25 is the ENDIF keyword paired with the second IF keyword. The second IF group is embedded within the first.
- 8/ Go to position 29 of the RCL section of PO array if (expression 4) tests 'false'. Position 24 is the ENDIF keyword paired with the third IF keyword. The third IF group is embedded within the second.

10/ Go to position 34 of the RCL section of PO array after execution of all codes following the 'true' test of the ELSEIF (expression 2). Position 34 is the ENDIF keyword paired with the first IF keyword.